



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,795	09/26/2003	Philip Rousselle	ADVENT007US2	9466
28722 7590 11/27/2007 BRACEWELL & PATTERSON, L.L.P. P.O. BOX 969 AUSTIN, TX 78767-0969			EXAMINER YIGDALL, MICHAEL J	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 11/27/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/672,795

Applicant(s)

ROUSSELLE, PHILIP

Examiner

Michael J. Yigdall

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 05 September 2007.
- 2a) ☒ This action is FINAL. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-25 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-25 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. This Office action is responsive to Applicant's submission filed on September 5, 2007.

Claims 1-25 are pending.

#### ***Response to Amendment***

2. The objections to the specification and to claims 11 and 19 have been withdrawn in view of Applicant's amendment.

3. The rejection of claims 19-23 under 35 U.S.C. 101 has been withdrawn in view of Applicant's amendment.

#### ***Response to Arguments***

4. Applicant's arguments have been fully considered but they are not persuasive.

With respect to claim 1, Applicant contends that Codella does not teach the limitation "wherein said first publish/subscribe topic is identified by a first property of said second distributed application component." Specifically, Applicant contends that Codella merely discloses the deployment descriptor properties for the input ports of a message bean, but does not disclose how to identify a publish/subscribe topic for publishing a request originating from a different message bean (remarks, page 10; Applicant's emphasis).

However, the examiner disagrees. Codella teaches publishing a request from a first message bean to a destination specified in an output port property (see, for example, column 13, lines 29-36). The output port property corresponds to an input port property of a second message bean that is a recipient of the request (see, for example, column 9, lines 34-46). The input port

Art Unit: 2192

property similarly identifies the destination that is the source of the request (see, for example, column 7, lines 30-39). Here, the term “destination” refers to a publish/subscribe topic (see, for example, column 15, lines 61-63). Thus, Codella teaches a property of a second message bean that identifies a publish/subscribe topic for publishing a request from a first message bean. In other words, Codella teaches the limitation “wherein said first publish/subscribe topic is identified by a first property of said second distributed application component.”

With respect to claim 2, Applicant contends that Codella does not disclose the limitation “wherein said first property is a type of said second distributed application component.”

Specifically, Applicant contends that Codella merely discloses the deployment descriptor properties for the input ports of a message bean, but provides no disclosure for identifying a publish/subscribe topic based on a type of message bean (remarks, page 10; Applicant’s emphasis).

However, the examiner disagrees. As noted above, the input port property of the second message bean identifies a publish/subscribe topic. Furthermore, the input port property is a type of the message bean, such as the “MessagePort” type or the derived “MessageInputPort” type (see, for example, column 19, lines 5-25). Thus, Codella teaches the limitation “wherein said first property is a type of said second distributed application component.”

With respect to claim 3, Applicant contends that Codella does not disclose the limitation “wherein said recipient is identified by a second property of said second distributed application component included within said request.” Specifically, Applicant contends that nothing in

Codella suggest that an identification property of the second component is included within the request generated by the first component (remarks, pages 10-11; Applicant's emphasis).

However, the examiner disagrees. Codella teaches message formats that define the fields included within the requests (see, for example, column 10, line 51 to column 11, line 5). The "Operation" field, for example, is a property included within the request that identifies the message bean that implements the requested interface (see, for example, column 11, lines 6-11). Thus, Codella teaches the limitation "wherein said recipient is identified by a second property of said second distributed application component included within said request."

With respect to claim 4, Applicant contends that Codella does not disclose the limitation "wherein said second property is a unique identifier of said second distributed application component." Specifically, Applicant contends that Codella does not disclose that a unique identifier is included within the request generated by the message bean (remarks, page 11; Applicant's emphasis).

However, the examiner disagrees. Codella discloses that a container "will extract the corresponding field from an incoming message and use it to identify the instance of the message bean to invoke" (column 13, lines 15-17). Thus, the field included within the incoming message is a unique identifier of the message bean. In other words, Codella teaches the limitation "wherein said second property is a unique identifier of said second distributed application component."

With respect to claim 5, Applicant contends that Codella does not disclose the limitation "wherein said first publish/subscribe reply topic is identified by a type of said first distributed

application component.” Specifically, Applicant contends that Codella merely discloses the deployment descriptor properties for the input ports of a message bean, but does not provide any disclosure for how to identify reply topics for subscribing a given message bean—let alone basing the identification on component type (remarks, page 11; Applicant’s emphasis).

However, the examiner disagrees. Codella teaches a reply-to destination that is specified in an output port property of a message bean (see, for example, column 13, lines 55-63). Again, the term “destination” refers to a publish/subscribe topic (see, for example, column 15, lines 61-63). Thus, Codella teaches a property of a message bean that identifies a publish/subscribe reply topic. Furthermore, the output port property is a type of the message bean, such as the “MessagePort” type or the derived “MessageOutputPort” type (see, for example, column 19, lines 5-16 and 26-35). Thus, Codella teaches the limitation “wherein said first publish/subscribe reply topic is identified by a type of said first distributed application component.”

With respect to claim 7, Applicant contends that Codella does not disclose the limitations “wherein said second publish/subscribe request topic is identified by a type of said first distributed application component” and “wherein said second publish/subscribe reply topic is identified by a type of said third distributed application component.” Specifically, Applicant contends that Codella merely discloses the deployment descriptor properties for the input ports of a message bean, but provides no disclosure for how to identify a publish/subscribe topic for which to subscribe a given message bean—let alone using component types (remarks, page 11).

However, the examiner disagrees. See the reasoning presented above with respect to claims 2 and 5.

With respect to claim 11, Applicant contends that Codella does not disclose “receiving a type of said first distributed application component, a name of said first distributed application component, a list of all other types of distributed application components that will send request or replies to said first distributed application component, and a list of all other types of distributed application components that will be receiving requests or replies from said first distributed application component.” Specifically, Applicant contends that Codella does not disclose that the type of any application component is specified while registering the first component (remarks, page 12; Applicant’s emphasis).

However, the examiner disagrees. Applicant acknowledges that the deployment descriptor of Codella specifies the names of the input and output ports of the message bean and the names of the methods associated with the ports (remarks, page 12). Codella further teaches that the deployment descriptor specifies, for example, references to JMS connection factories associated with the message source and message target (see, for example, column 7, lines 30-67). A JMS connection factory is an “object or class of objects” (column 7, line 41) associated with a message bean, which is to say a type of the message bean. Furthermore, the input and output port properties themselves are types of the message bean, such as noted above. Thus, Codella teaches that types of the message beans are specified while registering the first message bean.

With respect to claim 15, Applicant contends that Codella does not disclose creating a publisher or a subscription to a publish/subscribe topic based on a type of component or message bean (remarks, page 12; Applicant’s emphasis).

However, the examiner disagrees. Codella teaches creating a publisher in terms of creating a message proxy (see, for example, column 7, lines 3-24). Codella teaches creating a

subscription to a publish/subscribe topic in terms of creating a message listener (see, for example, column 17, lines 1-12). The message proxies and messages listeners are created based on classes of components (see, for example, column 7, lines 11-13 and column 17, lines 1-4), which is to say types of components.

With respect to claims 19-23, Applicant contends that Codella does not describe publish/subscribe topics that are based on component types (remarks, page 12; Applicant's emphasis).

However, the examiner disagrees. The claims recite "a publish/subscribe request topic for every type of distributed application component" and "a publish/subscribe reply topic for every type of distributed application component" (claim 19). The claims further recite "publish/subscribe request topics pertaining to all types of distributed application components ..." (claim 20) and "publish/subscribe reply topics pertaining to all types of distributed application components ..." (claim 21). The language of the claims does not indicate that the publish/subscribe topics are somehow based on particular types of distributed application components. Rather, a reasonable interpretation of the claims is that the publish/subscribe topics exist for "every" and "all" types of distributed application components, regardless of the particular types. Codella teaches such publish/subscribe topics, as noted above.

With respect to claims 24 and 25, Applicant contends that Codella does not disclose "translating a message formulated according to request/reply semantics into publish/subscribe communications." Specifically, Applicant contends that nothing in Codella suggests that a



Art Unit: 2192

message that was originally formulated according to request/reply semantics can be translated to publish/subscribe communications (remarks, page 13; Applicant's emphasis).

However, the examiner disagrees. Codella is directed to integrating object-oriented software with message-oriented middleware (see, for example, column 3, lines 47-67). As set forth in the Office action, Codella teaches messages formulated according to request/response semantics (see, for example, column 5, lines 51-55). Codella further teaches a message proxy that translates such messages into JMS messages (see, for example, column 13, lines 29-36). The JMS middleware implements publish/subscribe communications (see, for example, column 15, lines 61-63). Thus, Codella teaches "translating a message formulated according to request/reply semantics into publish/subscribe communications."

### *Claim Rejections - 35 USC § 102*

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-25 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,804,818 to Codella et al. (art of record, "Codella").

With respect to claim (original) 1, Codella discloses a method for facilitating communications between components of a distributed application (see, for example, column 3,

lines 47-67, which shows facilitating communications among message beans that are components of a distributed application) comprising the steps of:

receiving a request from a first distributed application component (see, for example, column 9, lines 11-18, which shows a message proxy receiving a request from a first message bean), wherein a recipient of said request is a second distributed application component (see, for example, column 9, lines 34-46, which further shows that a recipient of the request is a second message bean); and

publishing said request on a first publish/subscribe request topic (see, for example, column 13, lines 29-36, which shows a message proxy publishing the request to a first request destination, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic), wherein said first publish/subscribe topic is identified by a first property of said second distributed application component (see, for example, column 19, lines 37-54, which shows that a first property of the second message bean identifies the first request destination).

With respect to claim 2 (original), the rejection of claim 1 is incorporated, and Codella further discloses that said first property is a type of said second distributed application component (see, for example, column 19, lines 37-54, which shows that the first property is a type of the second message bean).

With respect to claim 3 (original), the rejection of claim 2 is incorporated, and Codella further discloses that said recipient is identified by a second property of said second distributed

Art Unit: 2192

application component included within said request (see, for example, column 11, lines 6-11, which shows that a second property of the second message bean identifies the recipient).

With respect to claim 4 (original), the rejection of claim 3 is incorporated, and Codella further discloses that said second property is a unique identifier of said second distributed application component (see, for example, column 13, lines 13-17, which shows that the second property is a unique identifier of the second message bean).

With respect to claim 5 (original), the rejection of claim 2 is incorporated, and Codella further discloses the steps of:

subscribing to a first publish/subscribe reply topic (see, for example, column 13, lines 55-63, which shows a message listener subscribing to a first reply destination, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic), wherein said first publish/subscribe reply topic is identified by a type of said first distributed application component (see, for example, column 19, lines 37-54, which shows that a type of the first message bean identifies the first reply destination);

forwarding a reply posted on said first publish/subscribe reply topic to said first distributed application component (see, for example, column 14, lines 12-20, which shows a message listener forwarding a reply to the first message bean).

With respect to claim 6 (original), the rejection of claim 5 is incorporated, and Codella further discloses that said reply is generated by said second distributed application component in response to said request (see, for example, column 9, lines 47-58, which shows that the second message bean generates the reply in response to the request).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Codella further discloses the steps of:

subscribing to a second publish/subscribe request topic (see, for example, column 16, lines 31-39, which shows a message listener subscribing to a second request destination, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic), wherein said second publish/subscribe request topic is identified by a type of said first distributed application component (see, for example, column 19, lines 37-54, which shows that a type of the first message bean identifies the second request destination);

forwarding a request posted on said second publish/subscribe request topic to said first distributed application component (see, for example, column 16, lines 31-39, which shows a message listener forwarding a request to the first message bean), wherein said request is generated by a third distributed application component (see, for example, column 9, lines 11-18, which shows that a third message bean generates the request);

receiving a reply from said first distributed application component (see, for example, column 15, lines 30-32 and 37-40, which shows a callback proxy receiving a reply from the first message bean), wherein a recipient of said reply is said third distributed application component (see, for example, column 15, lines 49-52, which further shows that a recipient of the reply is the third message bean); and

publishing said reply on a second publish/subscribe reply topic (see, for example, column 15, lines 40-44, which shows a callback proxy publishing the reply to a second reply destination, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic), wherein said second publish/subscribe reply topic is identified by a type of said third

distributed application component (see, for example, column 19, lines 37-54, which shows that a type of the third message bean identifies the second reply destination).

With respect to claim 8 (original), the rejection of claim 7 is incorporated, and Codella further discloses that said second and third distributed application components are the same distributed application component (see, for example, column 4, line 65 to column 5, line 14, which shows that the same message bean generates and is a recipient of requests).

With respect to claim 9 (original), the rejection of claim 7 is incorporated, and Codella further discloses the step of, prior to forwarding said request posted on said second publish/subscribe request topic, identifying that a recipient of said request posted on said second publish/subscribe request topic is either said first distributed application component or all distributed application components (see, for example, column 8, lines 29-37, which shows identifying that one or more message beans are recipients of the request).

With respect to claim 10 (original), the rejection of claim 7 is incorporated, and Codella further discloses the step of, sending a callback object to said first distributed application component with said request posted on said second publish/subscribe request topic (see, for example, column 15, lines 32-36, which shows sending a callback proxy to the first message bean with the request).

With respect to claim 11 (currently amended), the rejection of claim 1 is incorporated, and Codella further discloses the step of, registering said first distributed application component prior to receiving said request, wherein said step of registering comprises:

Art Unit: 2192

receiving a type of said first distributed application component, a name of said first distributed application component, a list of all other types of distributed application components that will send request or replies to said first distributed application component, and a list of all other types of distributed application components that will be receiving requests or replies from said first distributed application component (see, for example, column 7, lines 30-67, which shows registering the first message bean including receiving a type and a name of the first message bean and the message beans that will send and receive requests and replies to and from the first message bean).

With respect to claim 12 (original), the rejection of claim 11 is incorporated, and Codella further discloses that said step of registering further comprises:

receiving a callback object, wherein said callback object directs requests from other distributed application components to said first distributed application component (see, for example, column 15, lines 37-44, which shows receiving a callback proxy to direct requests to the first message bean).

With respect to claim 13 (original), the rejection of claim 12 is incorporated, and Codella further discloses the step of, invoking said callback object to deliver said request to said first distributed application component (see, for example, column 15, lines 32-36, which shows invoking the callback proxy).

With respect to claim 14 (original), the rejection of claim 11 is incorporated, and Codella further discloses that said step of registering further comprises:

sending a callback object to said first distributed application component (see, for example, column 9, lines 51-54, which shows sending a callback proxy to the first message bean).

With respect to claim 15 (original), the rejection of claim 11 is incorporated, and Codella further discloses that said step of registering further comprises:

creating a publisher on a publish/subscribe request topic of each of said other type of distributed application component receiving a request from said first distributed application component (see, for example, column 7, lines 3-24, which shows creating message proxies for a request destination of message beans receiving a request from the first message bean);

creating a publisher on a publish/subscribe reply topic of each of said other type of distributed application component types receiving a reply from said first distributed application component (see, for example, column 7, lines 3-24, which shows creating message proxies for a reply destination of message beans receiving a reply from the first message bean);

creating a subscription on a publish/subscribe request topic of said type of said first distributed application component (see, for example, column 17, lines 1-12, which shows creating a message listener for a request destination of the first message bean); and

creating a subscription on a publish/subscribe reply topic of said type of said first distributed application component (see, for example, column 17, lines 1-12, which shows creating a message listener for a reply destination of the first message bean).

With respect to claim 16 (original), the rejection of claim 15 is incorporated, and Codella further discloses that said subscription on a publish/subscribe request topic of said type of said

first distributed application component includes a filter that only accepts requests addressed to said first distributed application component or all distributed application components (see, for example, column 10, lines 42-50, which shows a message repository that filters requests to the message beans).

With respect to claim 17 (original), the rejection of claim 15 is incorporated, and Codella further discloses that said subscription on a publish/subscribe reply topic of said type of said first distributed application component includes a filter that only accepts replies addressed to said first distributed application component (see, for example, column 17, lines 12-21, which shows a map that filters replies to the first message bean).

With respect to claim 18 (original), the rejection of claim 1 is incorporated, and Codella further discloses that said request comprises one or more instructions directed toward a task to be performed by said second distributed application component (see, for example, column 5, lines 23-36, which shows that the request comprises instructions directed to a service or task that the second message bean performs).

With respect to claim 19 (currently amended), Codella discloses a system for facilitating request/reply communications among components of a distributed application (see, for example, column 3, lines 47-67, which shows facilitating communications among message beans that are components of a distributed application, and column 5, lines 51-55, which further shows request/response communications) comprising a distributed computing system, tangibly embodying thereon:



a publish/subscribe request topic for every type of distributed application component (see, for example, column 7, lines 48-67, which shows an output request destination for every message bean, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic);

a publish/subscribe reply topic for every type of distributed application component (see, for example, column 7, lines 30-47, which shows an input reply destination for every message bean, and column 15, lines 61-63, which further shows that the destination is a publish/subscribe topic); and

for each distributed application component,

a publisher on every publish/subscribe request topic within a portion of said publish/subscribe request topics (see, for example, column 9, lines 11-18, which shows message proxies for the destinations, and column 13, lines 29-36, which further shows that the message proxies are publishers);

a publisher on every publish/subscribe reply topic within a portion of said publish/subscribe reply topics (see, for example, column 9, lines 11-18, which shows message proxies for the destinations, and column 13, lines 29-36, which further shows that the message proxies are publishers);

a subscription on the publish/subscribe request topic pertaining to a type of said distributed application component (see, for example, column 17, lines 1-12, which shows a message listener for the destination pertaining to a type of the message bean); and

a subscription on the publish/subscribe reply topic pertaining to a type of said distributed application component (see, for example, column 17, lines 1-12, which shows a message listener for the destination pertaining to a type of the message bean).

With respect to claim 20 (original), the rejection of claim 19 is incorporated, and Codella further discloses that said portion of said publish/subscribe request topics includes publish/subscribe request topics pertaining to all types of distributed application components that receive requests from said distributed application component (see, for example, column 7, lines 3-24, which shows message proxies for destinations of message beans that receive requests).

With respect to claim 21 (original), the rejection of claim 19 is incorporated, and Codella further discloses that said portion of said publish/subscribe reply topics includes publish/subscribe reply topics pertaining to all types of distributed application components that receive replies from said distributed application component (see, for example, column 7, lines 3-24, which shows message proxies for destinations of message beans that receive replies).

With respect to claim 22 (original), the rejection of claim 19 is incorporated, and Codella further discloses:

one or more callback objects to facilitate delivery of requests and replies between said distributed application components and said publishers or subscriptions (see, for example, column 15, lines 37-44, which shows callback proxies to facilitate delivery of requests and replies among the message beans and the destinations).

With respect to claim 23 (original), the rejection of claim 22 is incorporated, and Codella further discloses:

routing logic to route a request or reply to a particular callback object (see, for example, column 15, lines 32-36, which shows routing logic to route requests or replies to a callback proxy).

With respect to claim 24 (original), Codella discloses a method of communicating messages between components of a distributed application (see, for example, column 3, lines 47-67, which shows communicating messages among message beans that are components of a distributed application) comprising the steps of:

receiving a message formulated according to request/reply semantics from a first distributed application component (see, for example, column 9, lines 11-18, which shows a message proxy receiving a message from a first message bean, and column 5, lines 51-55, which further shows that the message is formulated according to request/response semantics);

translating said message into publish/subscribe communications implemented by a publish/subscribe middleware program (see, for example, column 13, lines 29-36, which shows a message proxy translating the message into a JMS message, and column 15, lines 61-63, which further shows that the JMS middleware program implements publish/subscribe communications); and

forwarding said translated message to a second distributed application component (see, for example, column 9, lines 34-46, which shows a message proxy forwarding the translated message to a second message bean).

With respect to claim 25 (original), the rejection of claim 24 is incorporated, and Codella further discloses that said message is a request or reply (see, for example, column 4, line 65 to column 5, line 14, which shows that the message is a request or a reply).

### *Conclusion*

7. Applicant's amendment necessitated any new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY

Michael J. Yigdall  
Examiner  
Art Unit 2192

mjj



TUAN DAM  
SUPERVISORY PATENT EXAMINER